

The background of the entire page is a detailed, glowing blue circuit board pattern. It features a complex network of lines, pads, and circular components, resembling a high-density printed circuit board (PCB). The pattern is more prominent in the center and fades slightly towards the edges.

ROBOFUN.RO

LECȚIA I

CURS GRATUIT

ARDUINO ȘI ROBOTICĂ

Introducere în Arduino

Arduino Blink

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs

CC BY-NC-ND



Codul sursa din acest document este licentiat

Public-Domain

Esti liber sa distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, printare, sau orice alt mijloc), atat timp cat nu aduci nici un fel de modificari acestuia. Codul sursa din acest document poate fi utilizat in orice fel de scop, de natura comerciala sau nu, fara nici un fel de limitari.

Ce este Arduino ?

Arduino este una dintre cele mai simplu de utilizat platforme cu microcontroller. Te poti gandi la el ca la un minicalculator (are puterea de calcul a unui computer obisnuit de acum 15 ani), fiind capabil sa culeaga informatii din mediu si sa reactioneze la acestea.

In jurul lui Arduino exista un ecosistem de dispozitive extrem de bine dezvoltat. Orice fel de informatie ti-ai dori sa culegi din mediu, orice fel de conexiuni cu alte sisteme ai avea nevoie, exista o sansa foarte mare sa gasesti un dispozitiv pentru Arduino capabil sa iti ofere ceea ce ai nevoie.

Astfel, daca discutam despre preluarea de informatii din mediu, mai jos sunt doar cateva exemple de senzori : senzori ce determina nivelul de alcool in aerul respirat, senzor de incendiu, gaz GPL, monoxid de carbon, acceleratii ale dispozitivelor in miscare, curent consumat de diverse dispozitive casnice, forta de apasare, gradul de rotire, cartele RFID, distante, nivel de iluminare, directia nordului, prezenta umana, sunet, temperatura, umiditate, presiune atmosferica sau video. Daca ne referim la posibilitatea de a ne conecta cu alte sisteme, exista placi de retea Ethernet pentru Arduino capabile sa comunice informatii prin Internet, dispozitive capabile sa transmita date prin conexiune radio, placi de retea WIFI, dispozitive GSM pentru Arduino (capabile sa trimita / receptioneze SMS-uri, sa initieze apeluri de voce sau sa trimita date prin retea 3G) sau conectori Bluetooth pentru conectarea Arduino cu telefonul mobil sau laptop. In zona mecanica, exista motoare de curent continuu (utilizate pentru robotica), motoare pas cu pas (utilizate de obicei in zona industrială) sau servomotoare, controlate foarte exact. Pentru afisarea informatiilor preluate, exista ecrane LCD pentru Arduino, incepand cu cele mai simple (LCD text cu 16 caractere) pana la ecran LCD grafice. In prima parte a acestui curs vom dezvolta integrarea Arduino cu fiecare dintre dispozitivele pe care le-am amintit mai sus (si alte cateva in plus). In cea de-a doua parte a cursului vom prezenta pe larg o serie de proiecte interesante realizate prin combinarea Arduino cu unul sau mai multe dispozitive dintre cele enumerate mai devreme.

Pentru a-ti face o idee despre ceea ce se poate face cu Arduino, in continuare iti prezint cateva exemple de proiecte care se pot realiza cu Arduino (proiecte pe care le vom detalia pe larg in cea de-a doua parte a cursului) :

- senzor de alcool conectat la Arduino, Arduino conectat la PC; atunci cand nivelul de alcool in aerul expirat depaseste un anumit nivel prestabilit, computerul iti da un warning cand incerci sa postezi lucruri pe Facebook. Aici poti gasi [un articol cu mai multe poze despre acest proiect](#).
- robot autonom care ocoleste obstacole.

Cateva exemple de filme cu roboti care fac acest lucru :

- <http://www.youtube.com/watch?v=Xk3-HZpSm2U> - robot care parcheaza singur
- <http://www.youtube.com/watch?v=4nMNOugxf3I> - masina de taiat iarba (autonoma)
- <http://www.youtube.com/watch?v=X5nqCcMxGBs> - robot cu recunoastere video a obstacolelor
- <http://www.youtube.com/watch?v=6b4ZZQkcNEo> - robot dezvoltat de armata americana
- <http://www.youtube.com/watch?v=cNZPRsrwumQ> - un alt robot dezvoltat de armata americana

- robot controlat prin Bluetooth folosind telefonul mobil sau laptop-ul
 - aici poti gasi mai multe exemple de filme cu un astfel de robot <http://www.robofun.ro/magician-robot-arduino-driver-bluetooth>

- dispozitiv pentru pictat oua ("the EggBot").
 - <http://www.youtube.com/watch?v=ZC-VTRjaHk0>

- Arduino + senzor temperatura + senzor umiditate + senzor presiune atmosferica + placa de retea Ethernet care transmite datele de mediu pe Google Docs, la fiecare 10 secunde.

- mana robotica, bazata pe o manusa cu senzori de indoire si servomotoare.
 - aici gasesti un film cu acest gen de mana - <http://www.tehnorama.ro/animatronic-hand>

- Quadcopter-e
 - aici gasest un film cu mai multe quad-uri care zboara in formatie - <http://www.youtube.com/watch?v=YQIMGV5vtd4>

- [OpenEnergyMonitor](#), sistem bazat pe Arduino care monitorizeaza energia electrica consumata in casa.

- prastie reala (din lemn) combinata cu Arduino, un accelerometru si Bluetooth, pentru jucat Angry Birds pe PC.

– acces bazat pe cartele RFID + notificari prin Twitter.

În cazul Arduino, totul este foarte simplu de utilizat. Ai nevoie de circa 5 minute ca să îți instalezi mediul de dezvoltare și să scrii primul tău program pe Arduino. Nu este necesar să lipești fire, tot ce ai nevoie este un port USB liber.

Instalare Drive Arduino

Primul lucru pe care trebuie să îl faci înainte de a putea programa placa Arduino, este să îi instalezi driver-ele pe PC. Dacă rulezi Linux sau MAC, ai scapat, nu ai nevoie de driver-e. Dacă însă rulezi Windows, va trebui să instalezi și drivere-le, ca pentru orice alt dispozitiv hardware. Driver-ele sunt disponibile în mediul de dezvoltare, pe care îl vei descărca de la adresa <http://arduino.cc/en/Main/Software>. Descarcă arhiva .zip, și dezarchivează-o într-un director la alegerea ta. Conectează apoi placa Arduino la portul USB. PC-ul va detecta că ai un nou dispozitiv, și îți va cere să instalezi drivere pentru el. Alege opțiunea de instalare prin care îi spui Windows că ai tu drivere, și îi ceri să nu se conecteze la Internet pentru a căuta în mod automat. Selectează directorul "drivers" din directorul unde ai dezarchivat arhiva .zip de la pasul precedent (NU directorul "FTDI Drivers" !). Procesul continuă apoi cu "Next". În final, dacă deschizi Device Manager, în secțiunea "Ports (COM & LPT)" ar trebui să vezi o intrare numită "Arduino UNO".

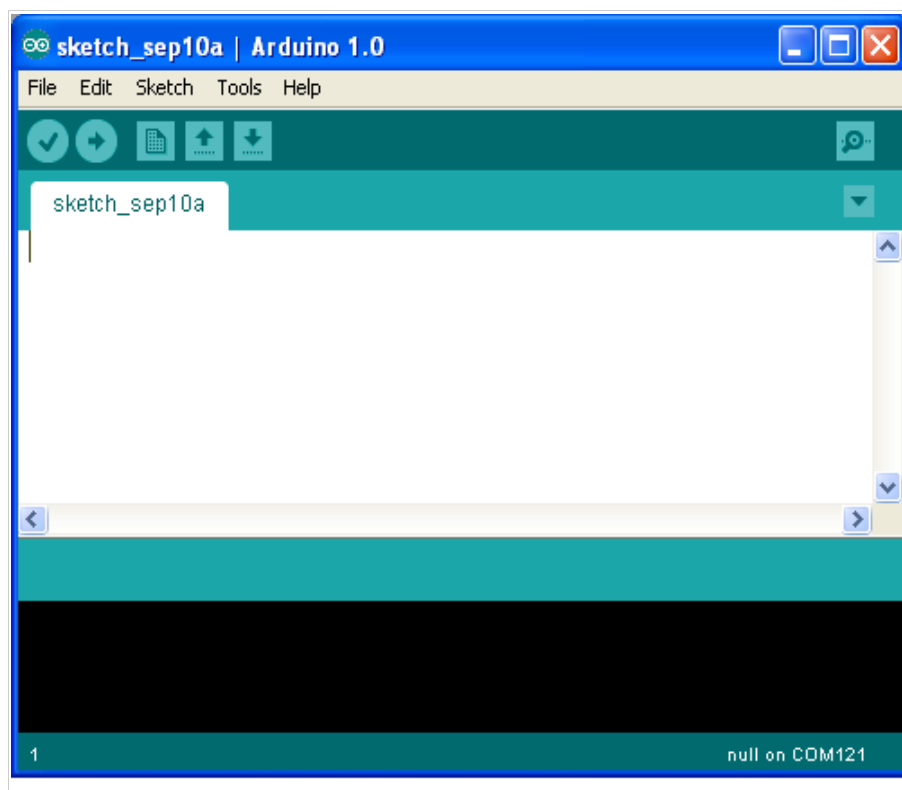
Aici - <http://arduino.cc/en/Guide/Windows> - găsești un ghid detaliat, cu screenshot-uri, referitor la instalarea driver-elor pentru Arduino.

Arduino IDE, Program Arduino, Upload

După ce ai instalat drivere-ele, următorul pas este să îți programezi placa Arduino. Rulează programul "arduino", din mediul de dezvoltare pe care tocmai l-ai descărcat la pasul precedent (vezi screenshot-ul de mai jos). Arduino se conectează la PC prin intermediul unui port serial. Primul pas pe care îl ai de făcut este să determini acest port. Cea mai simplă cale este de a conecta placa, aștepti circa 30 de secunde – 1 minut, ca să fii sigur că a fost detectată de PC, iar apoi deschizi meniul "Tools -> Serial Port". Ar trebui să vezi una sau mai multe intrări. Memorează-le (sau scrie-le pe o foaie de hartie / fă un screenshot). Deconectează placa Arduino din portul USB (scoate cablul din PC). Deschide din nou meniul "Tools -> Serial Port". Acel port care a dispărut este portul asociat plăcii Arduino. Conectează din nou placa Arduino la PC, așteaptă să fie recunoscută de PC, și apoi selectează portul respectiv din meniul "Tools -> Serial Port".

Următorul pas este selectarea tipului de placă cu care lucrezi. Din meniul "Tools -> Board", selectează tipul de placă cu care lucrezi (Arduino Uno,

Leonardo, Mega, etc).



Anatomia unui program Arduino

Orice program Arduino are doua sectiuni. Sectiunea "setup", care este rulata doar o singura data, atunci cand placa este alimentata (sau este apasat butonul "Reset"), si sectiunea "loop", care este rulata in ciclu, atat timp cat este alimentata placa. Sa luam un exemplu.

```
void setup() {  
    //codul scris aici ruleaza o singura data  
}  
void loop() {  
    //codul scris aici ruleaza tot timpul  
}
```

Astfel, in rutina "setup" vom pune de obicei cod de initializare, iar in rutina "loop" vom scrie partea principala a programului nostru.

Arduino Blink

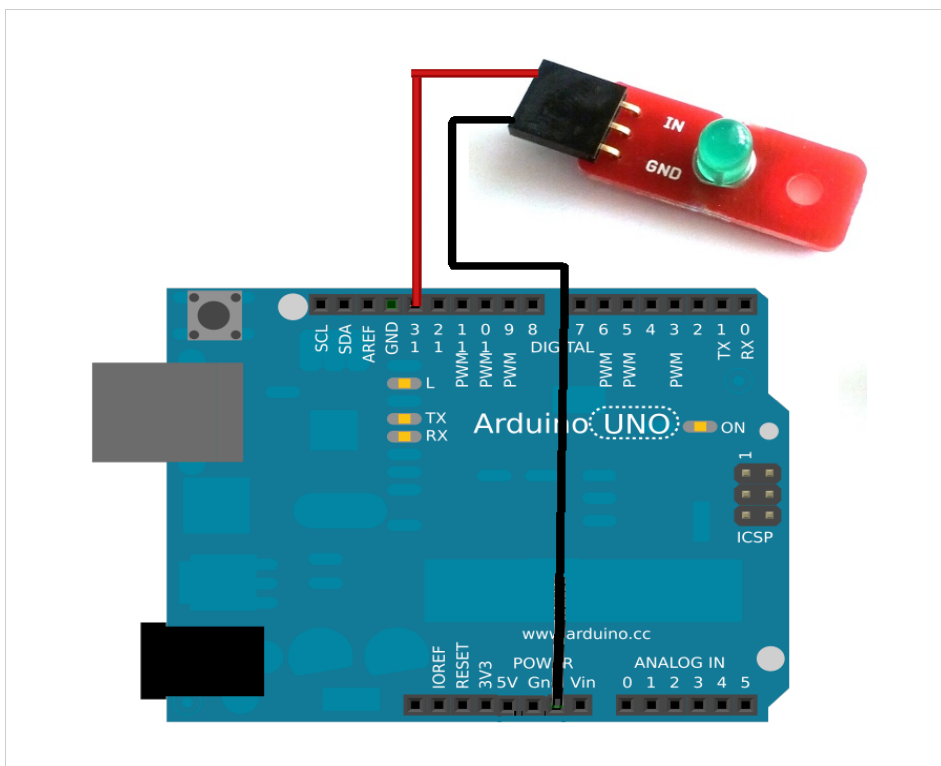
Pentru cel de-al doilea exemplu Arduino, vom folosi un led montat din fabricatie pe placa. Placa Arduino are din constructie un led conectat la pinul digital 13. Acest led se aprinde atunci cand pinul 13 digital este pus in HIGH (din programul scris pe Arduino) se stinge atunci cand pinul 13 este pus in LOW.

```
void setup() {  
    pinMode(13, OUTPUT);  
}  
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

Daca urci acest program pe placa Arduino si urmaresti led-ul conectat la pinul 13 vei observa ca el clipeste o data pe secunda. Sa analizam un pic programul de mai sus. In sectiunea "setup" avem o singura instructiune, care declara ca pinul 13 digital va fi folosit in continuare pentru a controla un dispozitiv (daca am fi vrut sa citim informatie din mediul - sa atasam un senzor la el - am fi scris "pinMode(13, INPUT)"). Aceasta instructiune se executa o singura data. Urmeaza sectiunea "loop", care aprinde led-ul conectat la pinul 13 ("digitalWrite(13, HIGH)"), asteapta o secunda, il stinge si iarasi asteapta o secunda. Asa cum am spus mai sus, instructiunile din sectiunea "loop" se executa ciclic, astfel ca imediat dupa ce s-a terminat un ciclu, led-ul iarasi se aprinde, apoi iarasi se stinge si tot asa pana cand taiem alimentarea placii. Exemplul urmator este extrem de similar cu acesta, doar ca in locul led-ului montat din fabrica pe placa Arduino vom utiliza un led fizic conectat la pinul 13 al placii.

Led Brick, ON/OFF

Un led brick reprezinta cea mai simpla posibilitate de a genera un semnal luminos folosind Arduino. Un led brick necesita un port digital liber pentru a fi controlat de catre Arduino.



Arduino GND	GND Led
Arduino Digital 13	IN Led

```
void setup() {  
    pinMode(13, OUTPUT);  
}  
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

Rutina *setup*, care se executa exact o singura data cand Arduino este alimentat, declara pinul digital 13 (cel la care am conectat led-ul) ca fiind un pin *de iesire* (in sensul ca va controla un dispozitiv extern conectat la Arduino, si nu va citi o informatie digitala din mediu).

Rutina *loop*, care se executa atata timp cat Arduino este alimentat, aprinde led-ul ridicand pinul 13 in HIGH (mai exact, face ca tensiunea pinului sa fie 5 V), apoi asteapta o secunda, stinge led-ul (tensiunea pinului devine 0 V) si apoi iarasi asteapta o secunda.

De ce nu merge ?

–sigur ai conectat corect firele ?

–sigur programul s-a incarcat pe Arduino (ai vazut mesajul "Done Uploading") ?

–daca scoti firul de conectare din pinul 13 (marcat pe led cu "IN") si il muti in pinul VCC, se aprinde led-ul ? (daca nu, atunci led-ul probabil este defect)

Aceasta a fost lectia 1. In final, as vrea sa te rog sa ne oferi feedback asupra acestei lectii, pentru a ne permite sa le facem mai bune pe urmatoarele.

Este vorba despre un sondaj cu 4 intrebari (oricare este optionala), pe care il poti accesa [dand click aici](#). Sondajul este complet anonim.

Sau ne poti contacta direct prin email la contact@robofun.ro .

Iti multumim,

Echipa [Robofun.RO](#)